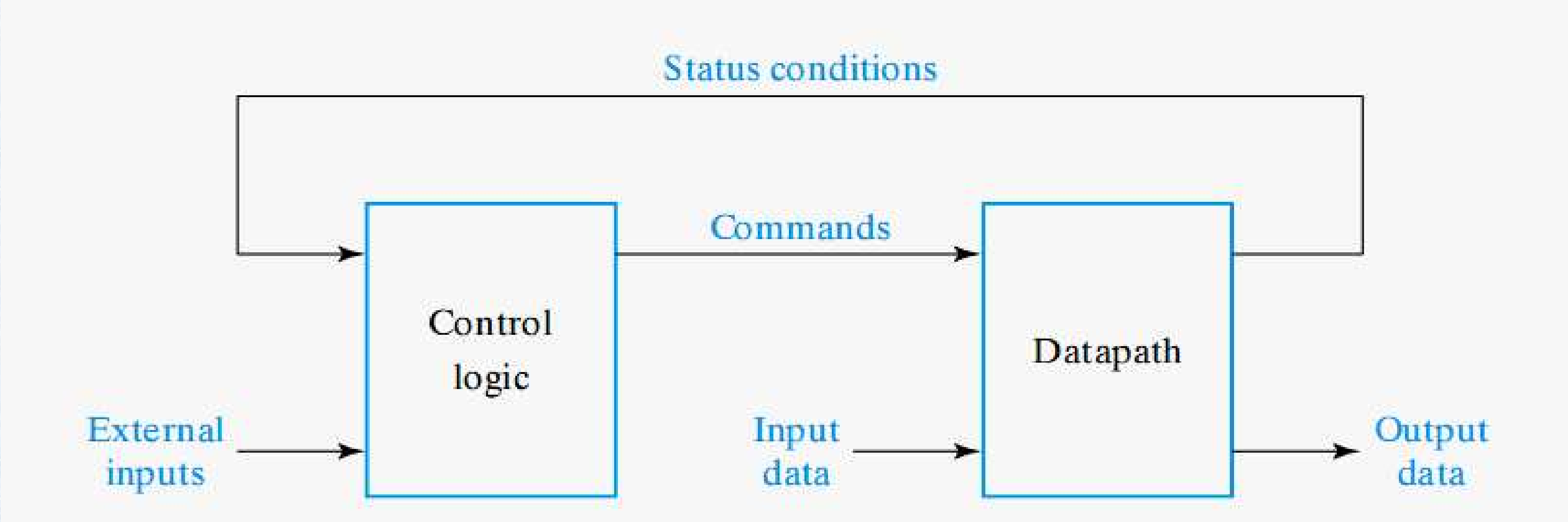
C hapter 5

ASM Chart

Introduction

- Any digital system can be thought of as decomposing of two major parts: the datapath part and the control logic part.
- The datapath part is responsible of the required operations by this digital system. Operations may be add, shift, increment and so on. The datapath may contain registers, flip-flops and any assisting logic gates. The control logic part constitutes the organizer of the required operations and responsible for



- To accomplish a certain operation, the control unit must issue one or more consecutive control signals to the datapath part organizing the usage of its parts.

- The feedback coming from datapath part may also initiate different sequence of control signals according to its value.

- In order to describe the sequence of operations required by a digital system we use a special type of charts called ASM (Algorithmic State Machine) chart.

ASM chart

- This type of chart is used to describe the sequence of operations required by a digital system. It is composed of three basic elements:

1- State box 2- Decision box 3- Conditional box

1- State box

- Used to indicate register operation

or output in the system.

. It contains the required register

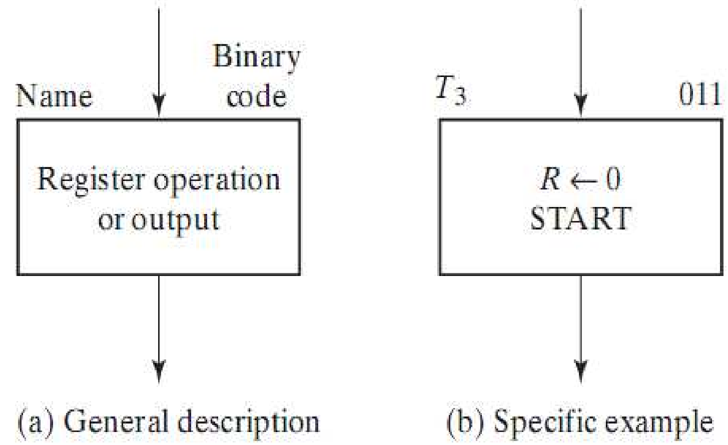
operation or output to be perforn

. On its upper corners, we will wri

the control signal associated with

one time as symbol and the other

time as binary code

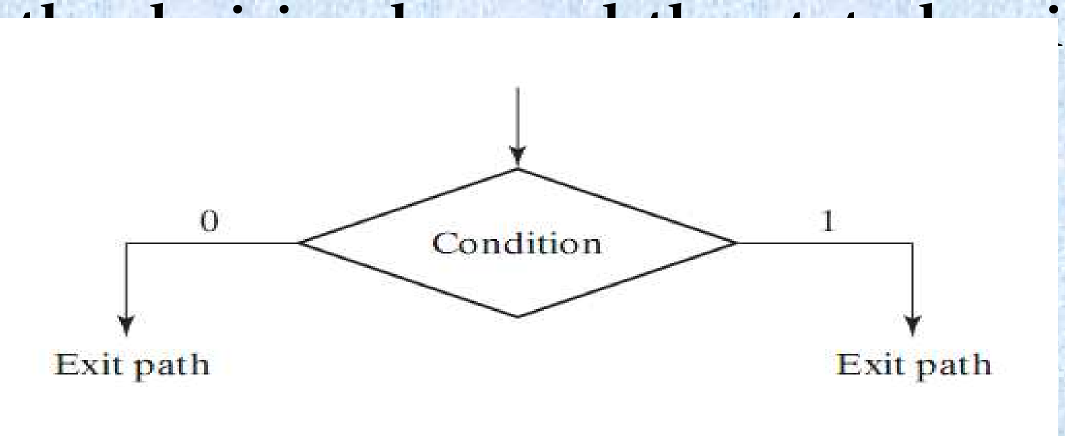


(a) General description

(b) Specific example

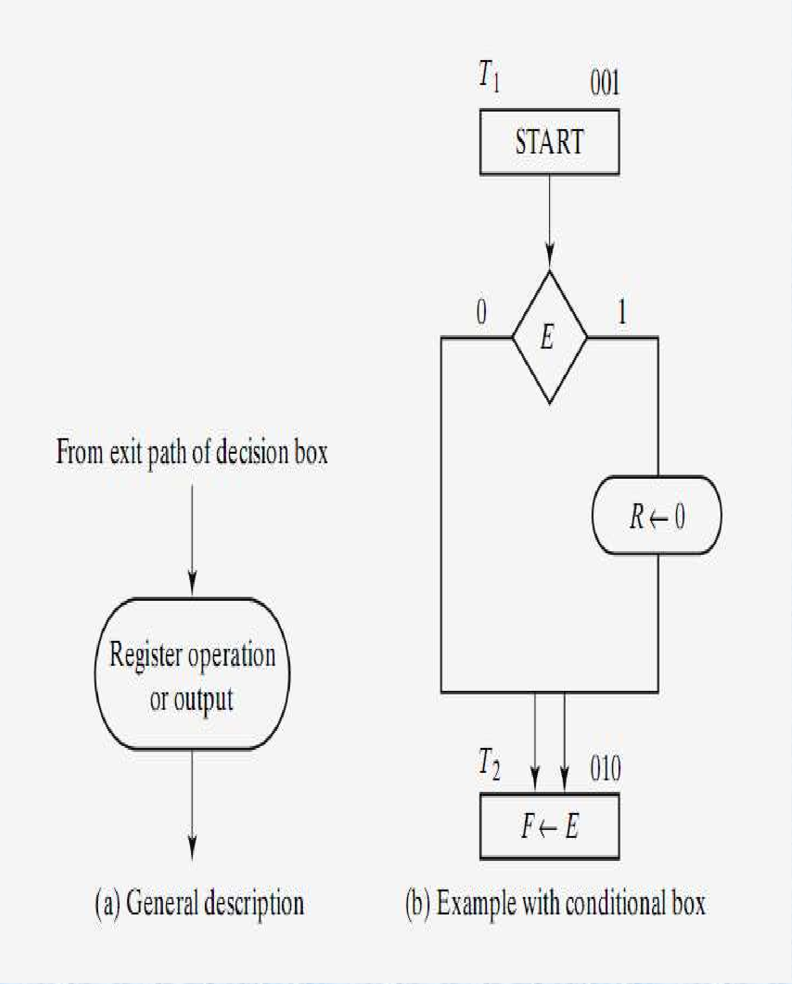
2- Decision box

- A decision box is used to indicate the branching to two or more different paths according to condition to be checked
- It is represented as a diamond box that contains the tested condition.
- It may have two or more paths as exit paths. If the condition has two possibilities, there will be two paths one for each possibility.
- We can write the value of the condition on the paths.
- One path is for 0 and the other for 1.
- We have used both in flowcharts before.



3- Conditional box

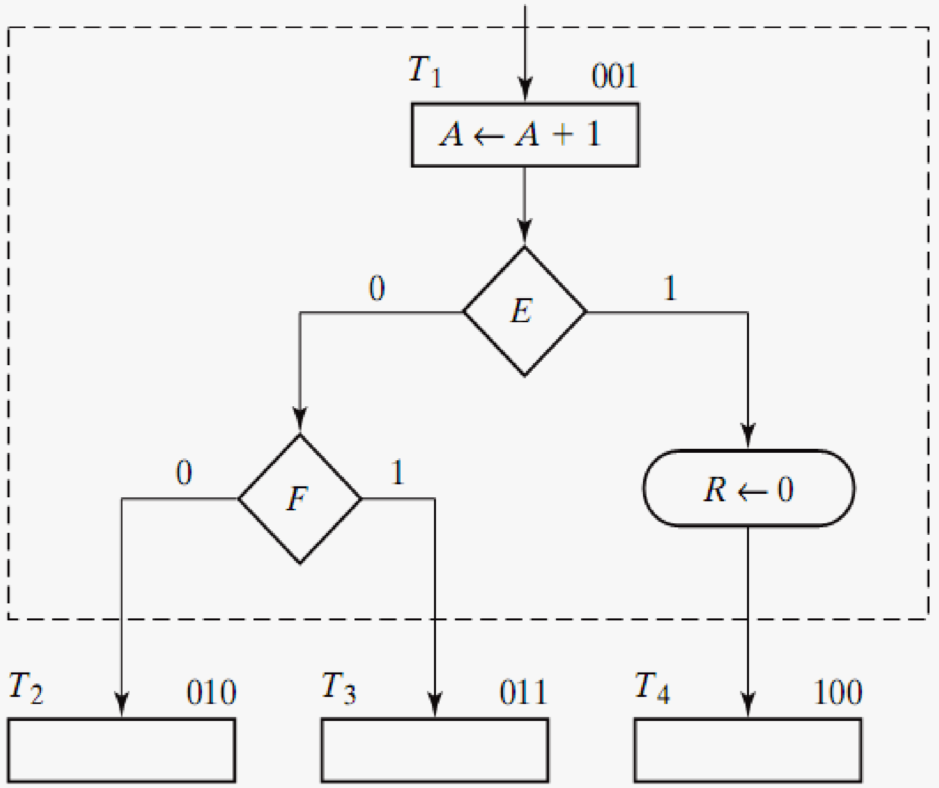
- It is represented as oval shape.
- As state box, it also contains register operation or output in the system but its input must come from a decision box.
- The held register operation or output inside it is executed when a certain value of input occurs and tested at the decision box.
- As an example, the value of E is tested at the decision box.
- If E = 1, the resetting of R is executed on the right branch while if E = 0, there is nothing to be done.
- This occurs at the same state of the above state box, i.e. it occurs in this

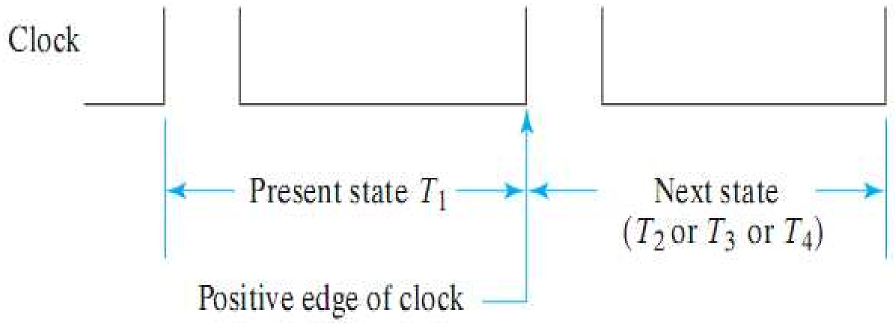


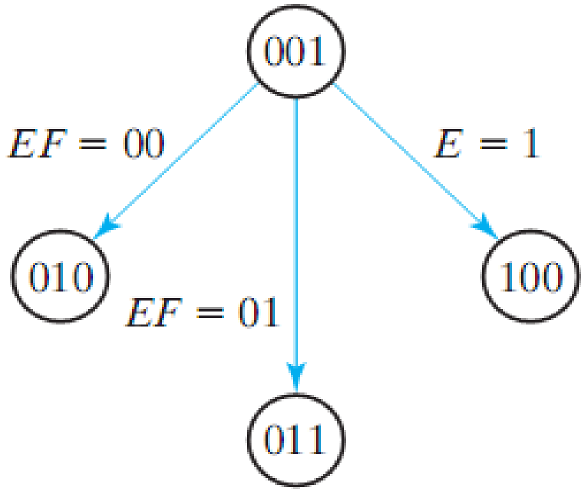
(a) General description

(b) Example with conditional box

- **Note that, the check in the decision box is also executed at state T1.**
- **The system moves to state T2 afterwards in both branches.**
- **We can thus determine a part of the chart that is executed at the same state. This part is called a block.**
- **A block contains a state followed by decision box and the branches containing conditional boxes until the beginning of the next state.**


T_1
001
$A \leftarrow A + 1$
0
E
1
0
F
1
$R \leftarrow 0$
T_2
010
T_3
011
T_4
100

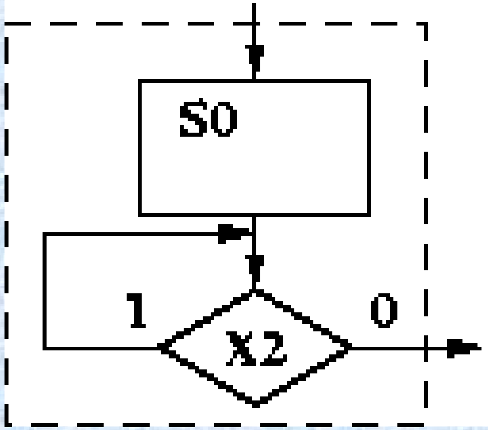

Clock
Present state T_1
Next state
(T_2 or T_3 or T_4)
Positive edge of clock

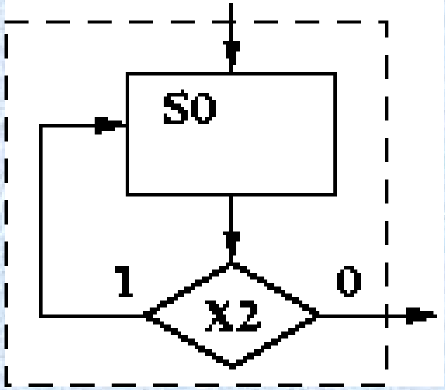

001
$EF = 00$
$E = 1$
010
$EF = 01$
100
011

Examples of register and flip-flop operations

Symbol	Meaning
$A \leftarrow B$	Move the contents of register B into register A
$A \leftarrow 0$	Reset the contents of register A
$A \leftarrow A + 1$	Increment the value stored in register A by 1
$A \leftarrow A - 1$	Decrement the value stored in register A by 1
$A \leftarrow A + B$	Add contents of register B to contents of register A and store the results in register A
$E \leftarrow 0$	Reset the content of flip-flop E

- No feedback internal to a state box is allowed. The following diagram indicates valid and invalid cases.



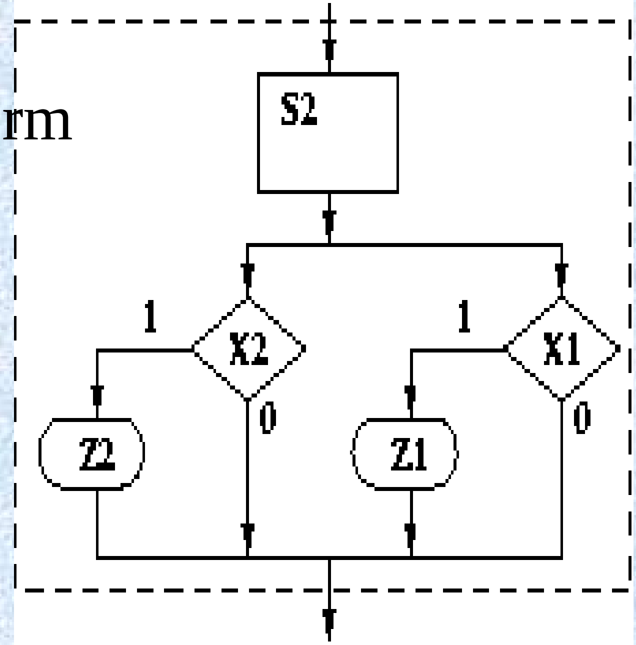


Incorrect **Correct**

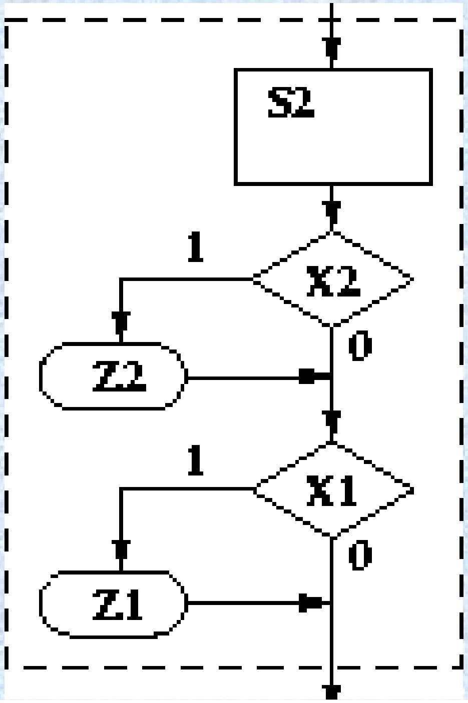
Parallel vs. Serial

- We can split the rules, several internal paths can be active, provided that they lead to a single exit path.
- Regardless of parallel or serial form, all tests are performed concurrently.
- Usually we prefer to use the serial form. The following two examples are equivalent.

Parallel Form

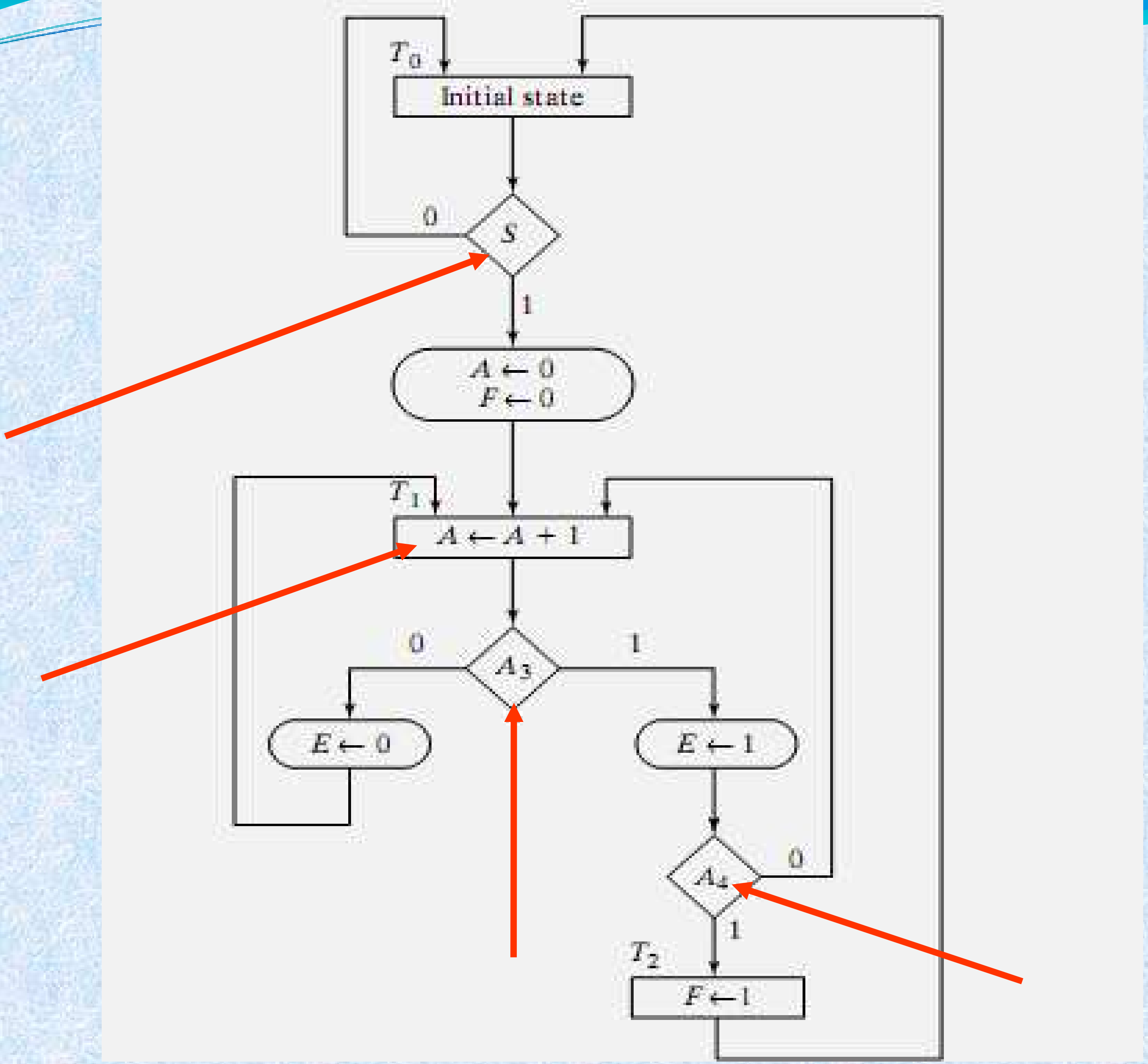


Serial Form



Design example(1)

- Design a digital system that contains: a counter and two flip-flops.
- The counter begins counting from 0 when a start signal comes.
- It then continues counting until the count reaches 0100, where E flip-flop is set to 1.
- The count continues after that until we reach 1100 where the second F flip-flop is set to 1 and the system stops counting and ends work.


T_0
Initial state
0
S
1
$A \leftarrow 0$
$F \leftarrow 0$
T_1
$A \leftarrow A + 1$
0
A_3
1
$E \leftarrow 0$
$E \leftarrow 1$
0
A_4
1
T_2
$F \leftarrow 1$

In the figure we have two ASM blocks.

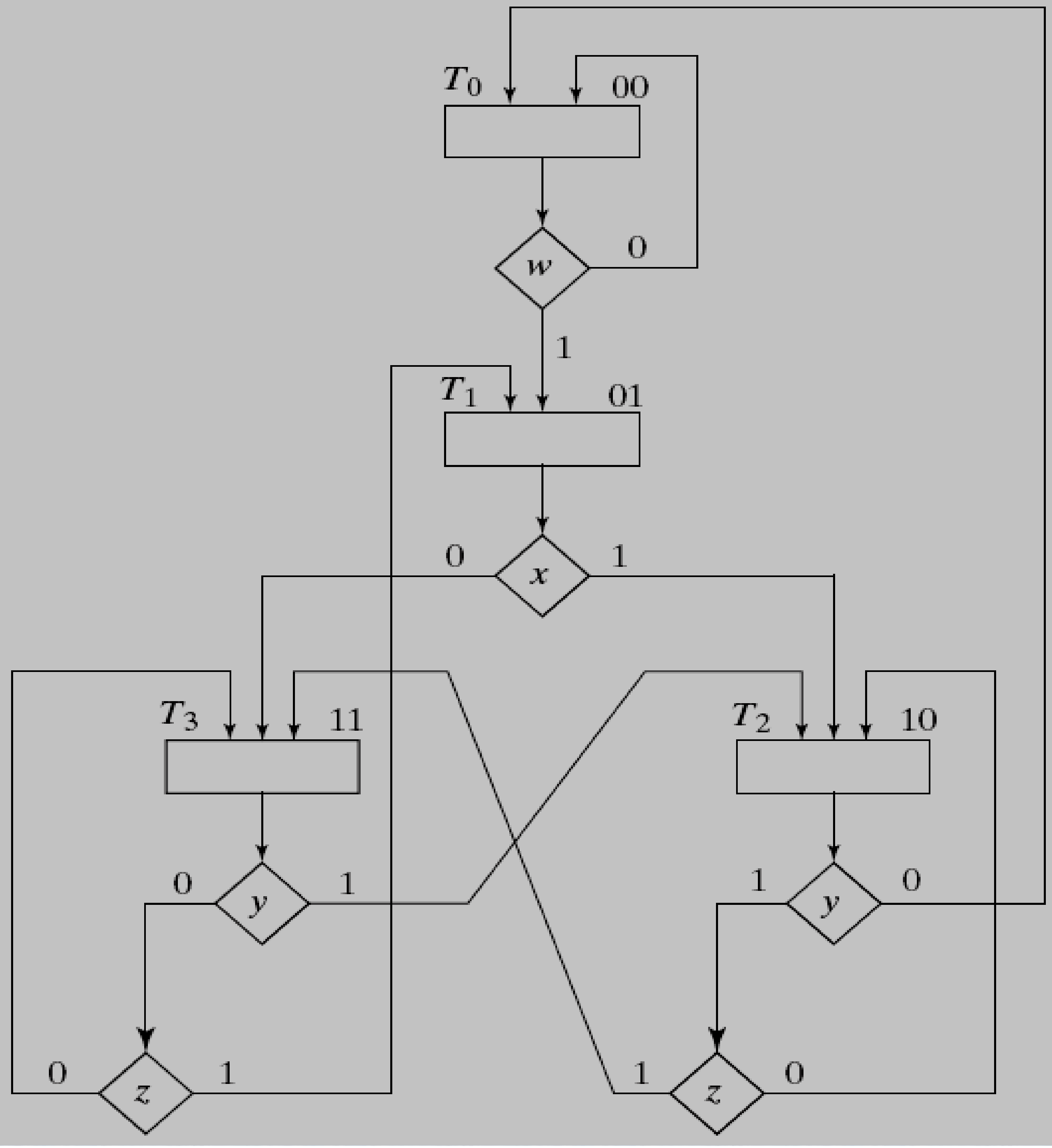
- The first one contains To state box, a decision box checking S signal and the conditional box resting counter A and flip-flop F.

- (**Put a rectangle on them on the above figure**).

- The second block contains T1 state box, two decision boxes checking A3 and A4 and two conditional boxes at the two branches coming from the checking of A3.

- (**Put a rectangle on them on the above figure**).

- Note that the actions contained in the same block occurs at the same clock pulse. (**important**).



Hall, Inc.

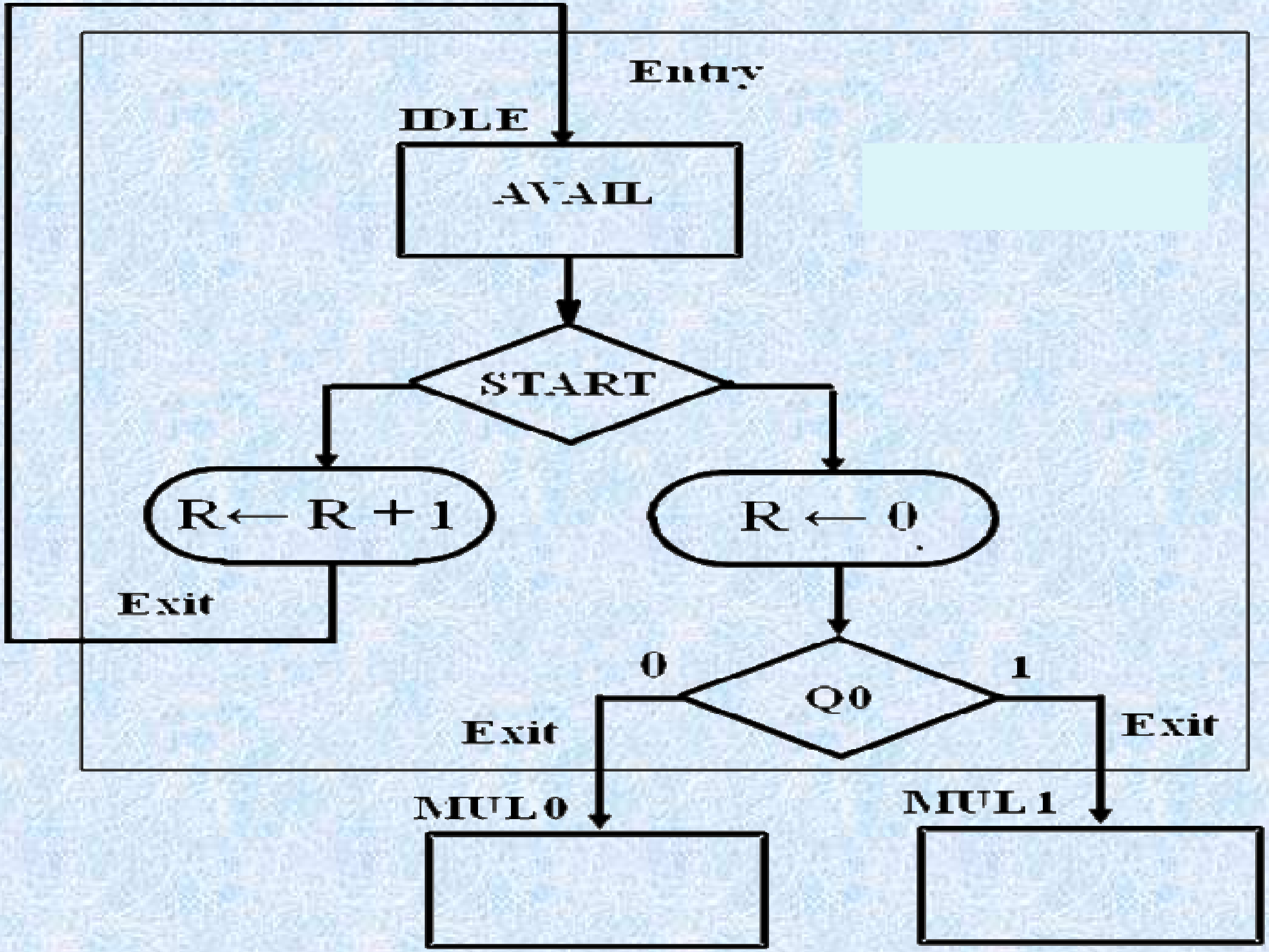
Quiz:

1- How many states in the chart, and what are they.

2-What are the register operations in the chart.

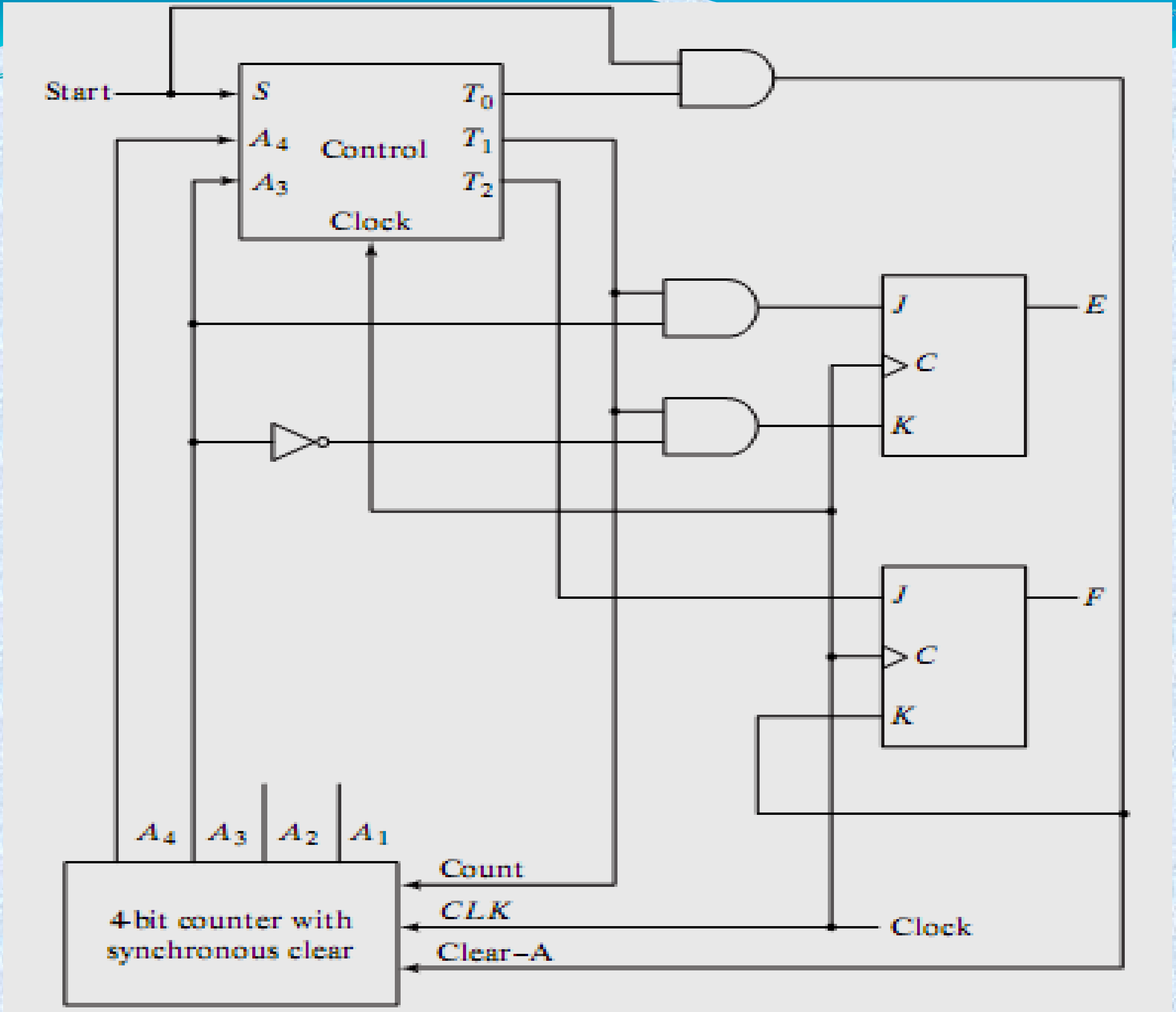
3- what are the values checked?

3-Identify the ASM block(s) in the diagram, and state to which state they belong?



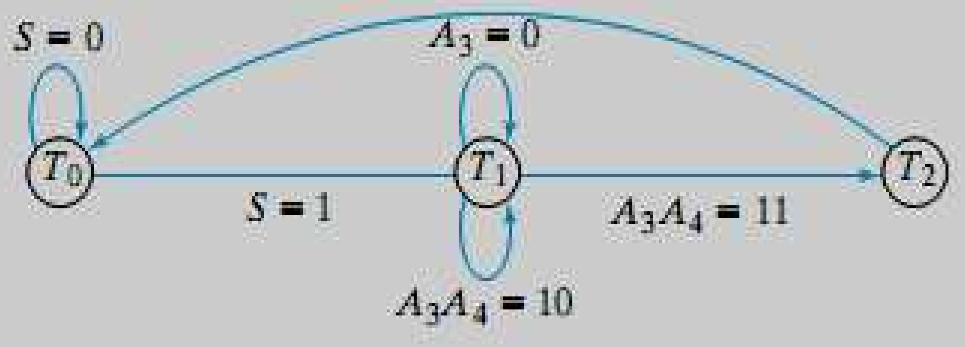
Designing the circuit

- **Such system must contain both control logic and datapath part.**
- **The datapath part will contain a 4-bit counter, two flip-flops E and F and additional gates to facilitate the setting and resetting of the flip-flops.**
- **We will choose the JK type flip-flop.**
- **The 4-bit counter will have the count input which orders it to increment the current count by 1, the clock input and the clear input which clears the value found in the counter and returns it to count 0.**


Start
S
A4
A3
Control
Clock
T0
T1
T2
J
C
K
E
J
C
K
F
A4
A3
A2
A1
4-bit counter with synchronous clear
Count
CLK
Clear–A
Clock

Control design

We begin the design of the control unit by drawing the state diagram for the control.



(a) State diagram for control

T_0: if ($S = 1$) then $A \leftarrow 0, F \leftarrow 0$

T_1: $A \leftarrow A + 1$

if ($A_3 = 1$) then E $\leftarrow 1$

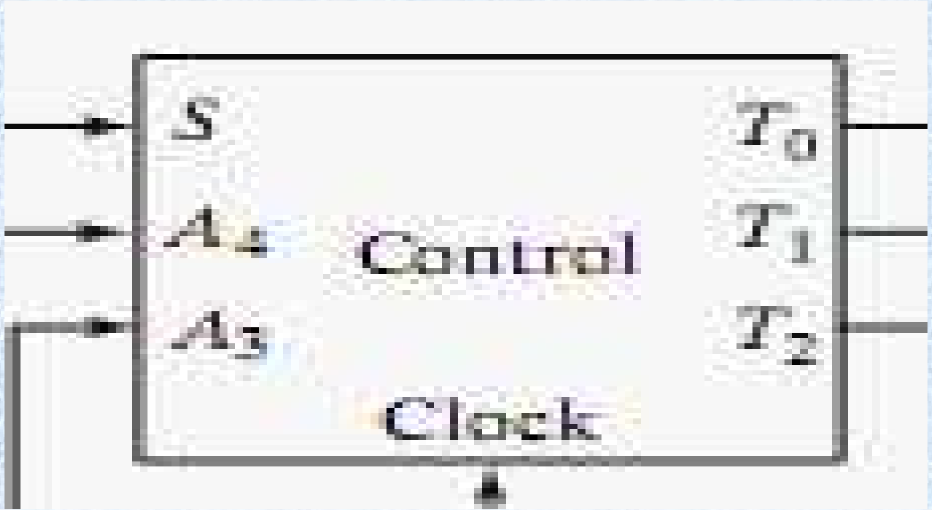
if ($A_3 = 0$) then E $\leftarrow 0$

T_2: $F \leftarrow 1$

(a) Register transfer operations

1- Design of control using JK Flip-flops

- To design the circuit with JK flip-flop, we will need
  Number of flip-flops = n where $(2)^n$ = number of states
- So, in this example, we need 2 flip-flops to cover the 3 states.
- Note that the two flip-flops (called G1 and G2) will give 4 states, so there will be one unused state.
- For such sequential circuit, We have 3 inputs: S, A3 and A4 and 3 outputs (T's). So, we have 32 rows in state table.



PS Symbol	PS		Inputs			NS		Outputs		
	G1	G2	S	A3	A4	G1	G2	T0	T1	T2
T0	0	0	0	X	X	0	0	1	0	0
T0	0	0	1	X	X	0	1	1	0	0
T1	0	1	X	0	X	0	1	0	1	0
T1	0	1	X	1	0	0	1	0	1	0
T1	0	1	X	1	1	1	1	0	1	0
T2	1	1	X	X	X	0	0	0	0	1

PS		NS		Flip-flop G1		Flip-flop G2	
G1	G2	G1	G2	JG1	KG1	JG2	KG2
0	0	0	0	0	X	0	X
0	0	0	1	0	X	1	X
0	1	0	1	0	X	X	0
0	1	0	1	0	X	X	0
0	1	1	1	1	X	X	0
1	1	0	0	X	1	X	1

- JG1 = G2 A3 A4 KG1 = 1
- JG2 = S KG2 = G1
- T0 = G'2
- T1 = G'1 G2
- T2 = G1

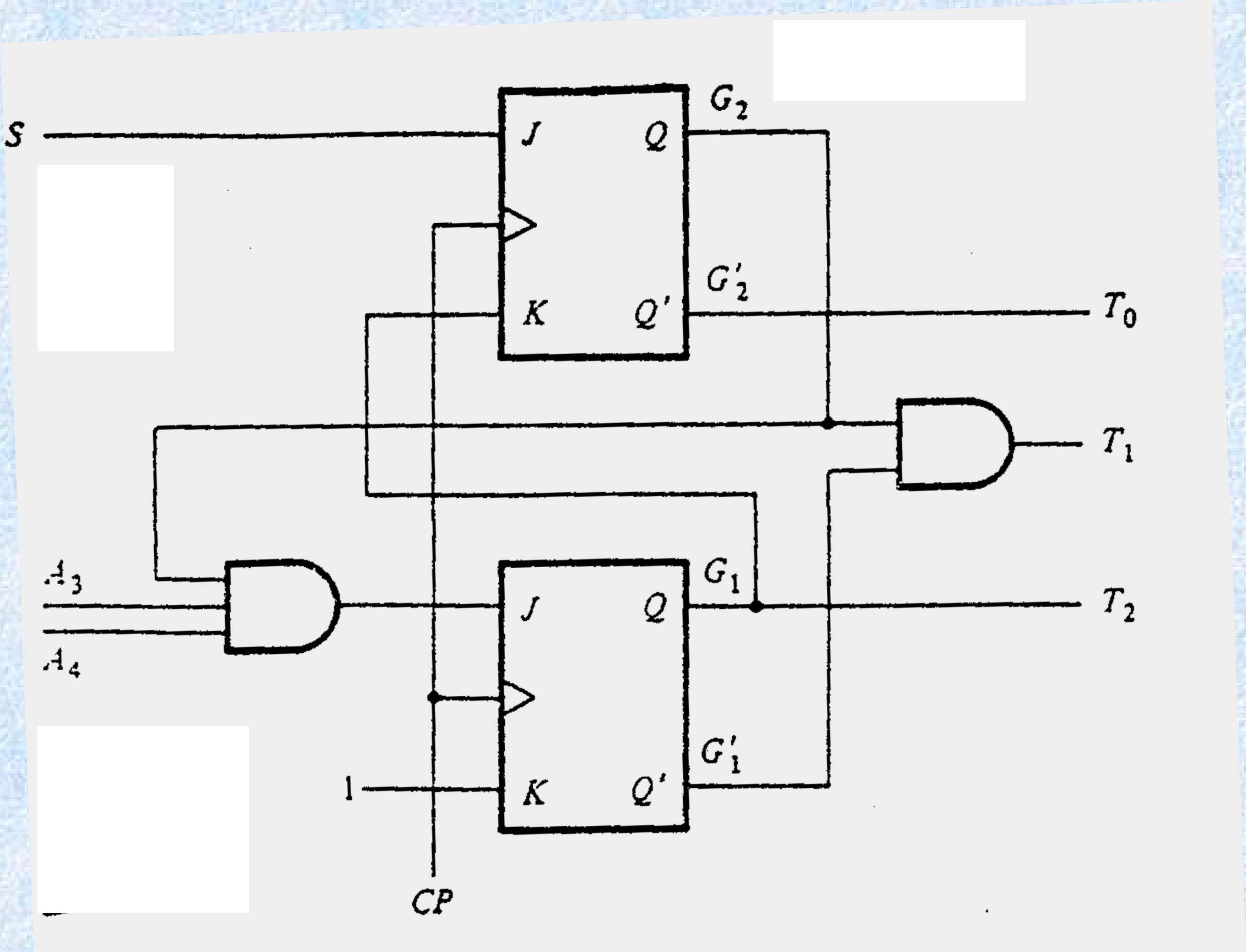


Figure 12 logic diagram of control using JK flip flop.

2- Design of control using D Flip-flops and decoder

- To design the circuit with D flip-flop, we will need

  Number of flip-flops = n where $(2)^n$ = number of states
- **So, We will need 2 flip-flops to cover the three states.**
- **Note that the two flip-flops will give 4 states so there will be one unused state.**
- **We will use a decoder this time to produce the outputs instead of logic circuit as above in the JK design.**
- **We use this method when the number of flip-flops and inputs in a state table is greater than five because it is very difficult to use maps in simplifications.**

PS Symbol	PS		Inputs			NS and D flip- flop inputs	
	G1	G2	S	A3	A4	G1 (D1)	G2 (D2)
T0	0	0	0	X	X	0	0
T0	0	0	1	X	X	0	1
T1	0	1	X	0	X	0	1
T1	0	1	X	1	0	0	1
T1	0	1	X	1	1	1	1
T2	1	1	X	X	X	0	0

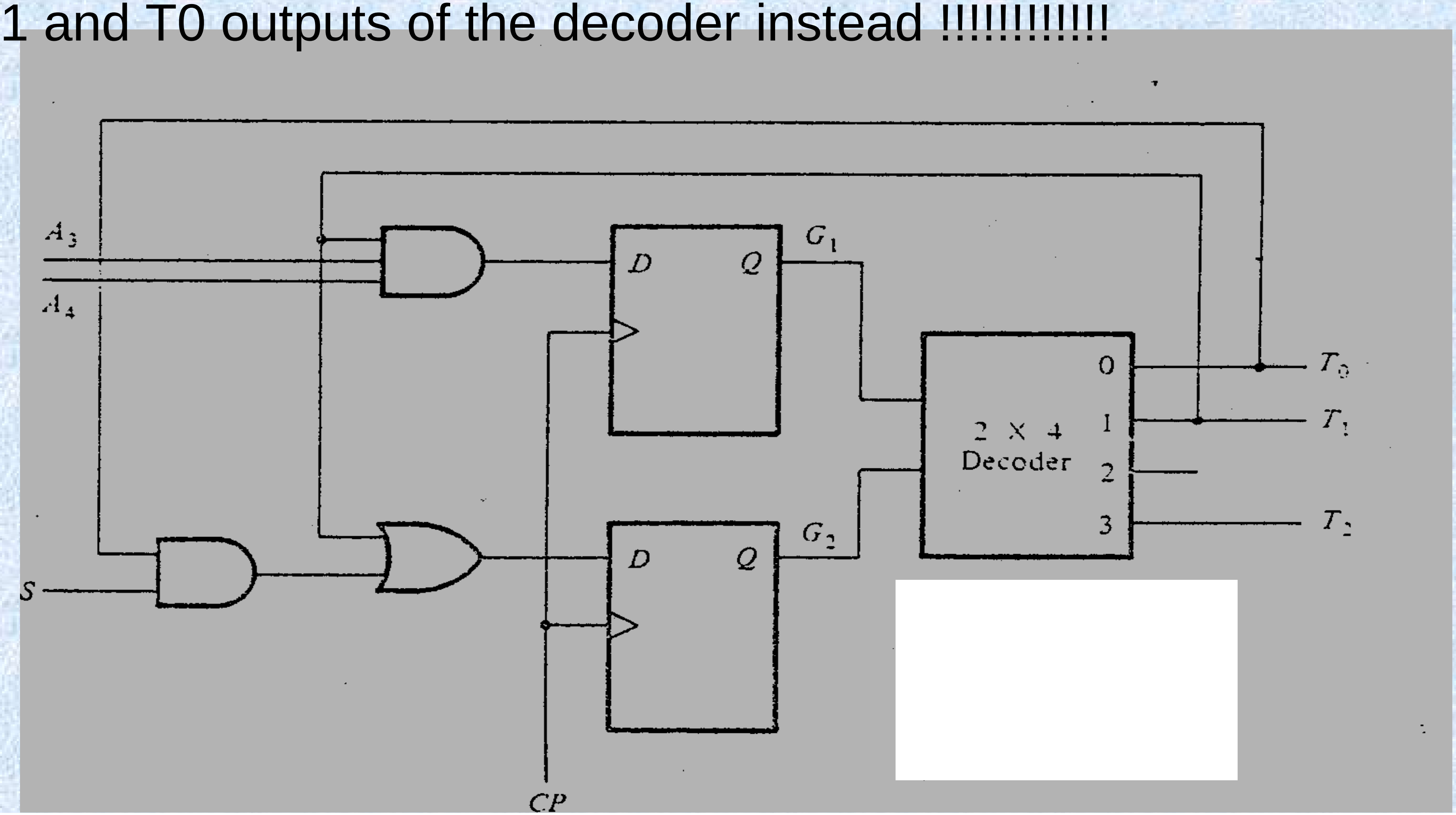
Simplifying the two columns of the D flip-flop inputs we get:

DG1 = G'1 G2 A3 A4

DG2 = G'1 G'2 S + G'1 G2

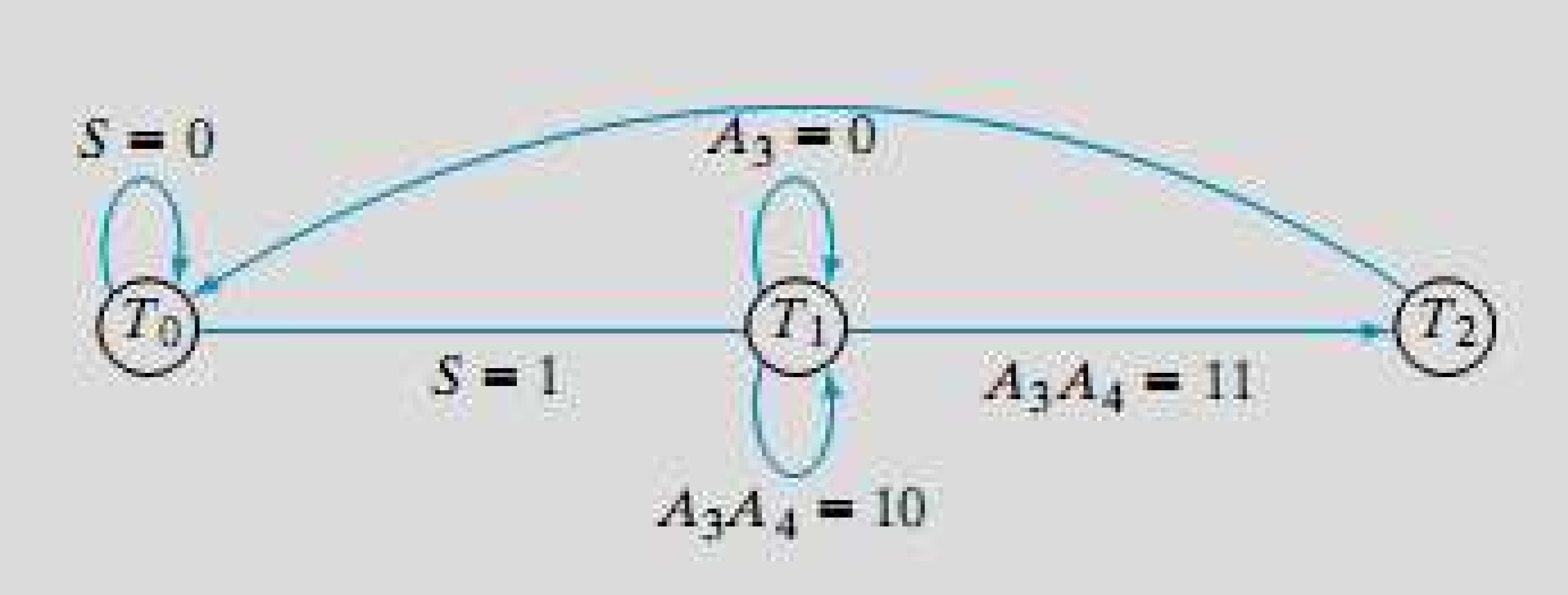
We will use a 2x4 decoder to produce the T state control signals. The inputs will be G1 and G2 and the outputs will be four states. We will use only three of them.

Instead of complicating the circuit to obtain the terms (G'1 G2 and G'1 G'2) found in the equations of DG1 and DG2, we will take T1 and T0 outputs of the decoder instead !!!!!!!!!!!!



3- Design of control using D Flip-flops only

- To design the circuit with D flip-flop only which is the simplest type of design as it is very straight forward in the design.
- We will need 3 flip-flops to cover the three states, one for each state.
- We will use the state diagram directly to produce the inputs of the D flip flops.
- The D flip-flop inputs will be deduced from the state diagram as the state possible transition cases.



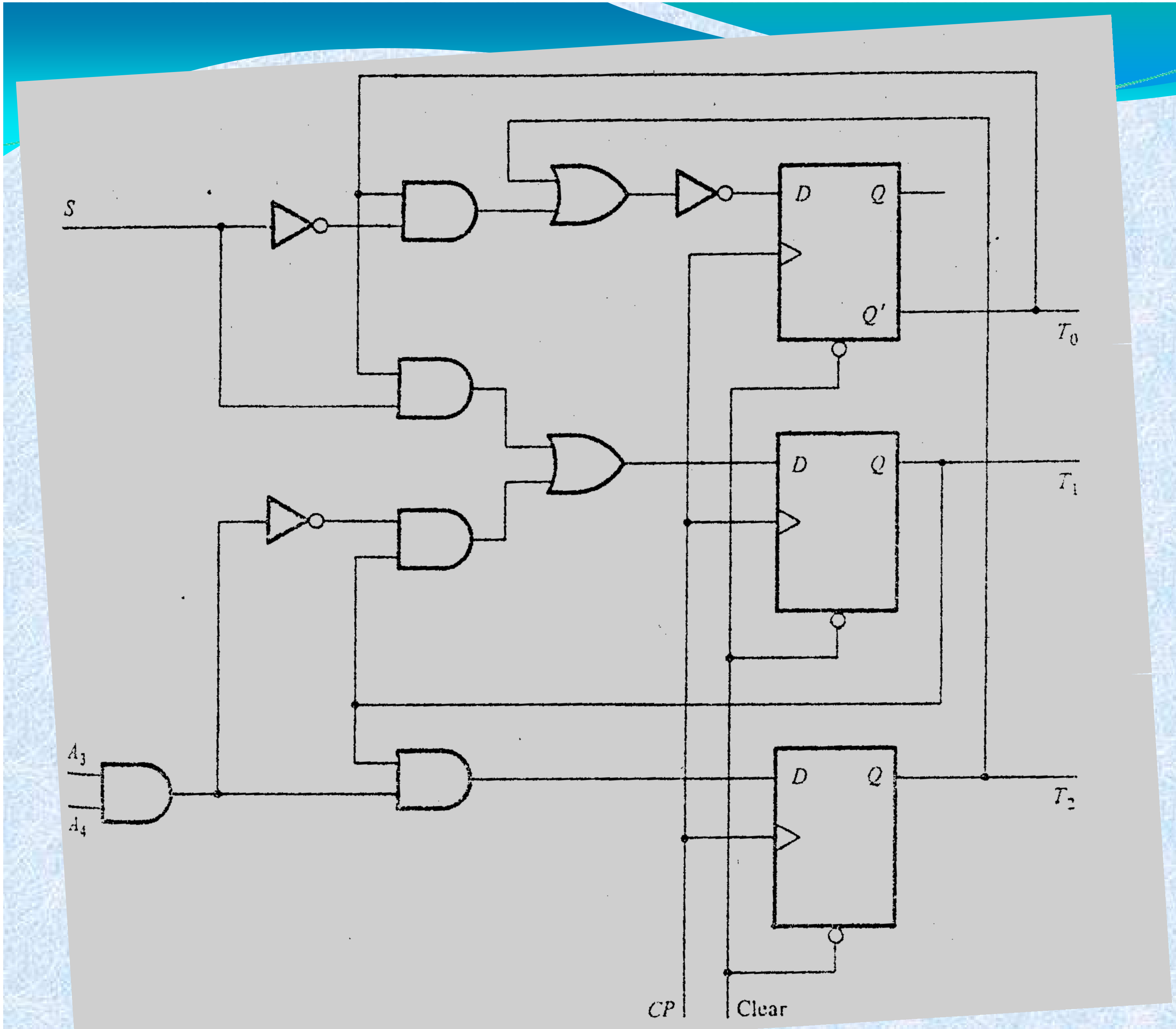
- For state To, the possible transitions will be To. S' and T2.

- For state T1, the possible transitions will be T1.A3 , To.S and T1.A3.A'4.

-

- For state T2, the possible transitions are T1. A3. A4.

- **(Draw the circuit diagram yourself)**


S
D
Q
Q'
T_0
D
Q
T_1
A_3
A_4
D
Q
T_2
CP
Clear